ARMY RESEARCH LABORATORY

Network Visualization Using Xj3D

by Andrew M. Neiderer

ARL-MR-754   **September 2010**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

Network Visualization Using Xj3D

Andrew M. Neiderer
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2010	Final	October 2009–January 2010

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Network Visualization Using Xj3D	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Andrew M. Neiderer	OTEDTC
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: RDRL-CII-C Aberdeen Proving Ground, MD 21005-5067	ARL-MR-754

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A solution for arbitrary viewing in three-dimensional space of data represented as a network (or graph) structure of nodes and link(s) is presented. Affine transformations of the data are followed by projection onto any plane using the Extensible 3D (X3D) standard application programming interface for manipulation and are then viewed in an Xj3D browser. X3D is an International Standards Organization standard for an extensible markup language description of scene content, which can be distributed across the Web and supports layering. In our case, text is displayed in a layer both statically and dynamically—a static display in the legend summarizes scene content, and a dynamic display in the console displays additional detail of any network node that is touched. A directed acyclic graph of X3D objects defines a scene. Both Java and ECMAScript bindings are used to access and manipulate the scene graph at run time.

15. SUBJECT TERMS

network, graph, visualization, X3D

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Andrew M. Neiderer
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)
Unclassified	Unclassified	Unclassified	UU	30	410-278-3203

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Ensuring national security may require that an individual's probable threat orientation be estimated. For example, security personnel at a checkpoint between country borders may need to quickly decide to detain, release, or release but watch a subject. An approach for these actions has been developed at the U.S. Army Research Laboratory (ARL) using a multidimensional scaling (MDS) technique.[1] MDS is a similarity analysis methodology for high-dimensional input, but is tolerant if the data is sparse and of varying reliability. This report does not address the internals of our MDS algorithm but does address the display of the resultant output.

An Extensible 3D (X3D) computer program was written at ARL for visualization of resultant MDS high-value individual (MDS-HVI) output. It has been labeled d^2NetVis—dynamic generation of nodes and links defining a dynamic network structure—and is written for the immersive profile, to include user interactivity (e.g., navigation) within a scene. Affine transformation(s) of data is/are followed by projection onto an arbitrary plane, which is sometimes referred to as a 2 1/2-D view. With minor modifications, the program can also be used for stereo viewing, i.e., truly three dimensional, using a Bitmanagement Software Contact[*] viewer.

The directed acyclic graph (DAG) of X3D objects is viewed in an Xj3D 2_M1_DEV_2008-05 -08 browser developed at Yumetech, Inc.[†] This open-source, stand-alone browser provides for some 170 X3D primitives, including an unlimited number of prototype definitions for a scene description.

2. A d^2NetVis Scene Graph Description

A d^2NetVis scene graph (SG) description of a network node has two branches: a geometric branch (red in figure 1) and a text branch (blue in figure 1). Both are a DAG of X3D objects, where the relationship among objects is a hierarchical parent-child structure with no cycling.[‡] In addition, the following X3D sensor and interpolator nodes are defined and attached to the SG:

---

[1] Hanratty, T.; Bornstein, A.; Brand, J.; Heilman, E.; Neiderer, A.; Richardson, J. Similarity Based Intelligence Data Fusion In The Human Battlespace; U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, draft report, September 2010.

[*] For additional information, go to http://www.bitmanagement.com/en/download/.

[†] The viewer can be found at http://www.xj3d.org/snapshots.html, and the functionality of an X3D object and its attribute(s) at http://www.web3d.org/x3d/content/X3dTooltips.html.

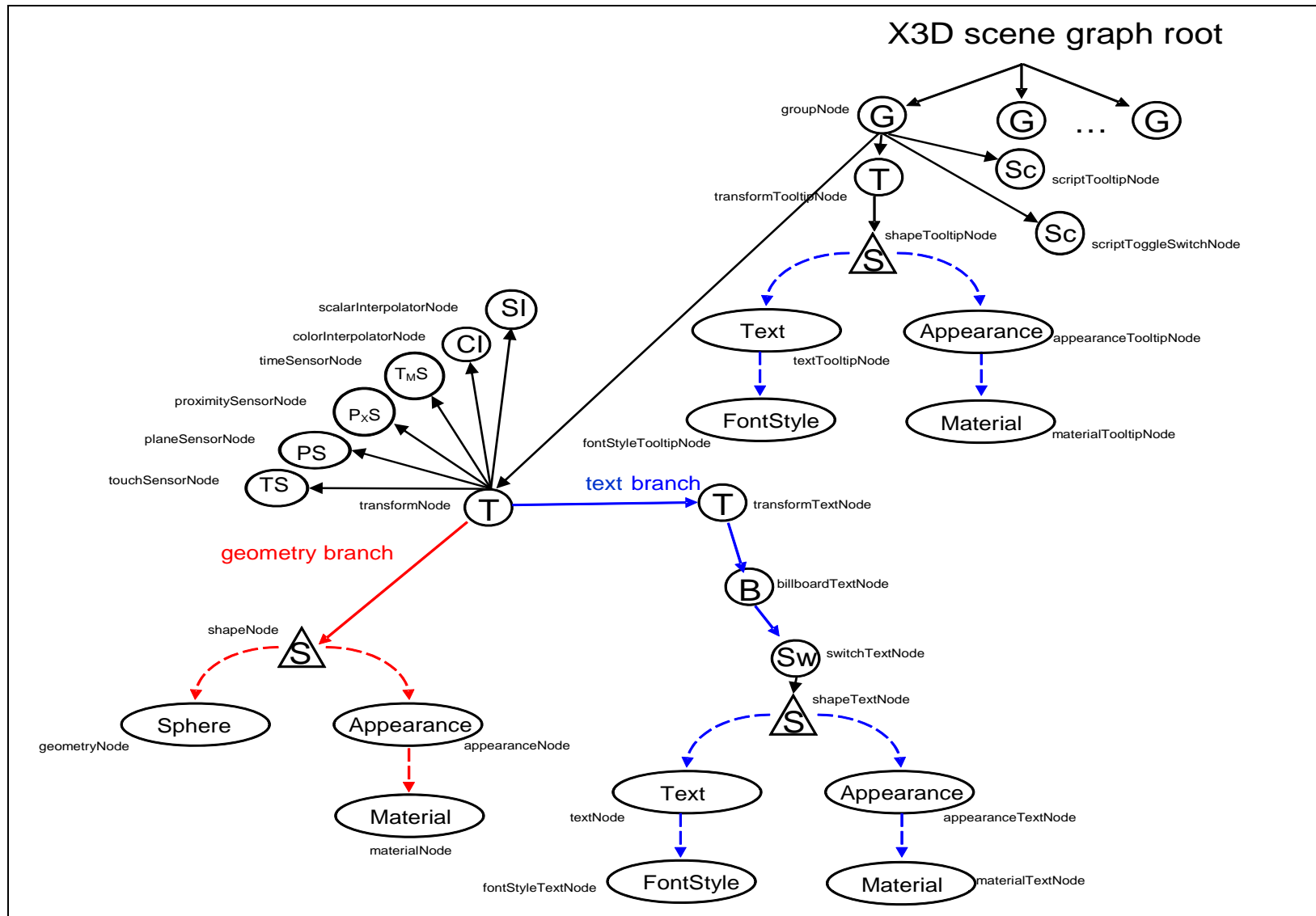[‡] Acyclic means that an object in the scene graph is not revisited.

Figure 1. DAG of X3D scene graph objects describing a network node (includes both geometric and text branches). Each network node has two such branches for a parent <Transform>, which is attached to a <Group>.

a <TouchSensor>, a <PlaneSensor>, a <ProximitySensor>, a <TimeSensor>, a <ColorInterpolator>, and a <ScalarInterpolator>. Finally, provision of two X3D <Script>s allows us to embed program code and use these sensors/interpolators to (1) toggle to display text and (2) provide for a tooltip capability.

A template for a network node is given in appendix A. For each network node to be added to the SG, this X3D code should be modified as follows before adding to the <Scene>:

1. The character string "RI", which refers to an individual of remote inquiry, should be changed (e.g., "C1" for criminal one).

2. The "translation" field of the root <Transform>, which is DEFed as "TRANSFORM_RI", needs to be updated with the appropriate x, y, z components of the current position vector.

3. The "offset" field of the <PlaneSensor>, which allows us to move a network node in 3-space and is necessary for link extrusion, is updated with the x, y, z components.

After a unique character string and location for each network node are specified, next update the <ROUTE>s (see appendix B for the <ROUTE> template).

X3D defines a mechanism for routing events across these X3D sensor, interpolator, and script nodes. In d^{2}NetVis, an event cascade is defined for each network node as follows:

1. Sending a text event, which is animated, to the console

2. A tooltip

3. Movement in 3-space

Once this information is provided for each node to be added, the X3D scene is live and can be manipulated.

3. An Example MDS-HVI Visualization

The following d^{2}NetVis view (see figure 2) is an example of a 53-node network resulting from MDS-HVI application. The summary of scene content is located in a legend layer at the left of the window and consists of the following: a reference individual (RI), 5 criminals (C), 10 innocents (I), 8 militia (M), 10 non-STEF (NS), 12 STEF (S), and 7 STEF hostages (SH). A tooltip capability allows one to pass the mouse pointer over any network node for quick identification (e.g., the M4 node). When a node is touched, i.e., mouse button 1 or 2 is pressed when over the network node, a more detailed text message (e.g., M4 is militant-4) is displayed in the console layer at the bottom of the window.
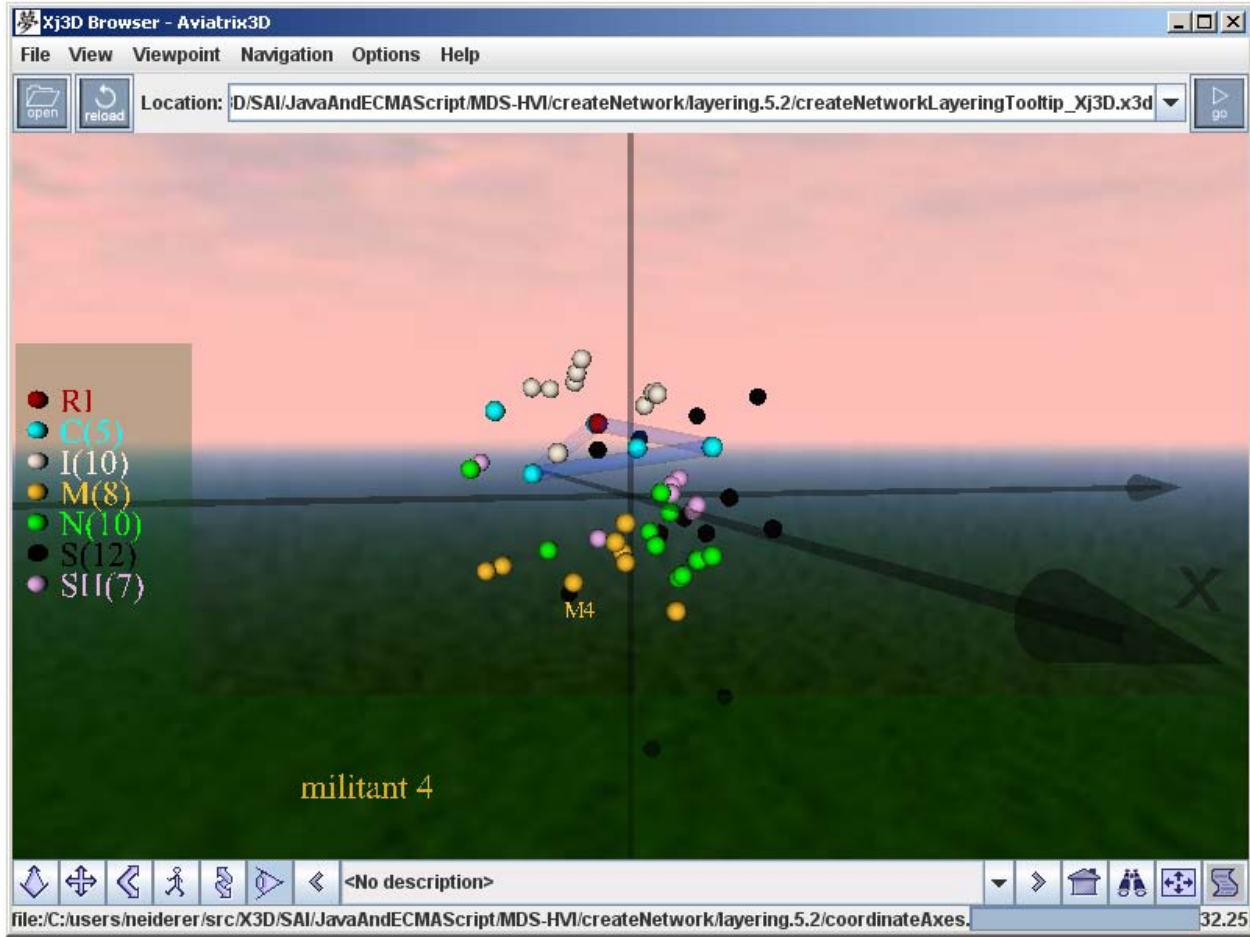
Figure 2. An Xj3D view of a 53-node network with a legend (at left) and a console (bottom). Note the tooltip display "M4." When this network node is touched, it results in the text "militant 4" being animated in the console for 10 s.

4. Conclusions and Future Work

X3D objects necessary for navigation within a d^2NetVis scene have been identified (see figure 1) and discussed in this report. We then provided for link extrusion between nodes, which may be necessary for an interactive environment (see figure 2, where a cyclic path exists between RI and two of the Cs). To satisfy this additional capability, an SG for the link(s) was then added to d^2NetVis. Each link is made dynamic, i.e., dynamic link extrusion, by defining a <Script> (e.g., see appendix C, SCRIPT_link_RI_C1); three <field>s in this <Script>, which correspond to affine transformations (translation, rotation, and scaling), are <ROUTE>d to the root <Transform>.

4

This SG is quite similar to the one for network nodes (see figure 1) except that the geometry is now a <Cylinder> instead of a <Sphere>. A supporting <PlaneSensor> was necessary for arbitrary extrusion of the <Cylinder>.

INTENTIONALLY LEFT BLANK.

Appendix A. Template for a Network Node

The following X3D code is a template that can be used for adding an individual node to a d²NetVis scene. Changes necessary are designated in red, which include (1) a unique descriptive character string and (2) 3D specification of node position.

```xml
<!-- 3D geometrical object(s), ie <Layer3D>, now follow; -->
<!--  RI, C(s), I(s), M(s), N(s), S(s), and SH(s).            -->

<!-- reference individual (note that now     -->
<!--  an ECMAScript SAI to 3D content).  -->

<Transform DEF="TRANSFORM_RI"
             translation="-0.0739 0.1408 0.0453">
 <Shape>
  <Appearance>
   <Material USE="MATERIAL_RI"/>
  </Appearance>

  <Sphere radius="0.01875"/>
 </Shape>

 <Transform DEF="TRANSFORM_tooltipHolder1RI">
  <Transform DEF="TRANSFORM_tooltip1RI"
             translation="-0.0 -0.0 0.0">
   <Billboard>
     <Shape>
      <Appearance>
       <Material USE="MATERIAL_RI"/>
      </Appearance>

      <Text DEF="TEXT_tooltip1RI"
           string="">
       <FontStyle size="0.1"/>
      </Text>
     </Shape>
   </Billboard>
  </Transform>
 </Transform>

 <Transform DEF="TRANSFORM_tooltipHolder2RI">
  <Transform DEF="TRANSFORM_tooltip2RI"
             translation="-0.0225 -0.0225 0.0225">
   <Billboard>
     <Shape>
      <Appearance>
       <Material USE="MATERIAL_RI"/>
```

```
        </Appearance>

        <Text DEF="TEXT_tooltip2RI"
                string="">
         <FontStyle size="0.05"/>
        </Text>
       </Shape>
     </Billboard>
    </Transform>
  </Transform>

  <!-- sensors and interpolators for animation -->

  <TouchSensor DEF="TOUCH_SENSOR_RI"
                enabled="true"
                description="RI (click to toggle details in console)"/>

  <PlaneSensor DEF="PLANE_SENSOR_RI"
                offset="-0.0739 0.1408 0.0453"/>

  <ProximitySensor DEF="PROXIMITY_SENSOR_RI"
                   size="10000.0 10000.0 10000.0"/>

  <TimeSensor  DEF="TIME_SENSOR_RI"
                enabled="true"
                loop="false"
                startTime="0.0"
                cycleInterval="10.0"/>

  <ColorInterpolator  DEF="COLOR_INTERPOLATOR_RI"
                    key="0.0 0.33 0.67 1.0"
                    keyValue="0.4 0.4 0.4, 1.0 0.5 0.5, 1.0 0.5 0.5, 0.4 0.4 0.4"/>

  <ScalarInterpolator DEF="SCALAR_INTERPOLATOR_RI"
                    key="0.0 0.5 1.0"
                    keyValue="1.0 0.0 1.0"/>
</Transform>

<!-- toggle display of text on the geometry, ie Sphere, in the console -->

<Script DEF="SCRIPT_toggleSwitchConsoleRI"
        directOutput="true">
  <field name="switchNodeRI" type="SFNode"
                               accessType="initializeOnly">
   <Switch USE="SWITCH_2DconsoleTextRI"/>
  </field>
```

```
  <field name="toggleRI"     type="SFTime"
                             accessType="inputOnly"/>

  <field name="descriptionRI"  type=     "MFString"
                                accessType="outputOnly"/>

  <![CDATA[ecmascript:
              function initialize()
              {
                  Browser.println('initialize() switchNodeRI: ' +
                                  switchNodeRI);
              }

              function toggleRI(dummy)
              {
                Browser.println('toggleRI() on entry, switchNodeRI.whichChoice: ' +
                                switchNodeRI.whichChoice);

                if ( switchNodeRI.whichChoice < 0 ) {
                 switchNodeRI.whichChoice = 0;
                }
                else {
                 switchNodeRI.whichChoice = -1;
                }

                descriptionRI = new MFString('reference individual' +
' – leader of Mashdan cell');

                  Browser.println('switchNodeRI: ' +
                                  switchNodeRI);
                  Browser.println('toggleRI() on exit, switchNodeRI.whichChoice: ' +
                                  switchNodeRI.whichChoice);
              }
    ]]>
  </Script>

  <Script DEF="SCRIPT_tooltipRI"
        directOutput="true">
   <field name="touchSensor"  type="SFNode"
                                accessType="initializeOnly">
    <TouchSensor USE="TOUCH_SENSOR_RI"/>
   </field>
   <field name="tooltip"     type="SFNode"
                             accessType="initializeOnly">
    <Transform USE="TRANSFORM_tooltip1RI"/>
   </field>
```

```
<field name="tooltipText"  type="SFNode"
                           accessType="initializeOnly">
  <Text USE="TEXT_tooltip2RI"/>
</field>
<field name="pointChanged" type="SFVec3f"
                           accessType="inputOnly"/>
<field name="overChanged"  type="SFBool"
                           accessType="inputOnly"/>
<field name="orientation"    type="SFRotation"
                           accessType="inputOutput"/>

<![CDATA[ecmascript:
                function initialize()
                {
                }

                function pointChanged(point)
                {
                 var a = orientation.angle;
                 var c = Math.cos(a);
                 var s = Math.sin(a);
                 var t = 1 - c;

                 var x = orientation.x;
                 var y = orientation.y;
                 var z = orientation.z;

                 var x1 = (t * x * x + c);
                 var x2 = (t * x * y + s * z);
                 var x3 = (t * x * z - s * y);
                 var y1 = (t * x * y - s * z);
                 var y2 = (t * y * y + c);
                 var y3 = (t * y * z + s * x);
                 var z1 = (t * x * z + s * y);
                 var z2 = (t * y * z - s * x);
                 var z3 = (t * z * z + c);

                 var newX = x1 * point.x + x2 * point.y + x3 * point.z;
                 var newY = y1 * point.x + y2 * point.y + y3 * point.z;
                 var newZ = z1 * point.x + z2 * point.y + z3 * point.z;

                 tooltip.translation.x = newX * 0.11;
                 tooltip.translation.y = newY * 0.11;
                }

                function overChanged(over)
```

```
                {
                 if ( over ) {
                  tooltipText.string = new MFString('RI');
                 }
                 else {
                  tooltipText.string = new MFString();
                 }
                }
        ]]>
  </Script>
```

Appendix B. Template for <ROUTE>ing a Network Node

This appendix appears in its original form, without editorial change.

The following X3D code is a template for adding <ROUTE>s to d^{2}NetVis scene, and makes use of existing network nodes (see appendix A). Changes necessary are designated in red, which correspond to a descriptive character string.

```
<!-- animation -->

<ROUTE fromNode="TOUCH_SENSOR_RI"                fromField ="touchTime"
       toNode   ="SCRIPT_toggleSwitchConsoleRI"  toField   ="toggleRI"/>
<ROUTE fromNode="SCRIPT_toggleSwitchConsoleRI"   fromField ="descriptionRI"
       toNode   ="TEXT_2DconsoleTextRI"          toField   ="string"/>


<ROUTE fromNode="TOUCH_SENSOR_RI"                fromField= "touchTime"
       toNode   = "TIME_SENSOR_RI"               toField   = "startTime"/>
<ROUTE fromNode="TIME_SENSOR_RI"                 fromField= "fraction_change!
       toNode   = "SCALAR_INTERPOLATOR_RI"       toField   = "set_fraction"/>
<ROUTE fromNode="SCALAR_INTERPOLATOR_RI"         fromField= "value_changed"
       toNode   = "MATERIAL_2DconsoleTextRI"     toField   ="transparency"/>


<ROUTE fromNode="TOUCH_SENSOR_RI"                fromField ="hitPoint_changed"
       toNode   = "SCRIPT_tooltipRI"             toField   ="pointChanged"/>
<ROUTE fromNode="TOUCH_SENSOR_RI"                fromField ="isOver"
       toNode   ="SCRIPT_tooltipRI"              toField   ="overChanged"/>
<ROUTE fromNode="PROXIMITY_SENSOR_RI"            fromField ="position_changed"
       toNode   = "TRANSFORM_tooltipHolder1RI"   toField   ="set_translation"
<ROUTE fromNode="PROXIMITY_SENSOR_RI"            fromField ="orientation_changed"
       toNode   = "TRANSFORM_tooltipHolder1RI"   toField   ="set_rotation"/>
<ROUTE fromNode="PROXIMITY_SENSOR_RI"            fromField ="orientation_changed"
       toNode   ="SCRIPT_tooltipRI"              toField   ="orientation"/>


<ROUTE fromNode="PLANE_SENSOR_RI"                fromField="translation_changed"
       toNode   ="TRANSFORM_RI"                  toField   ="translation"/>
```

Appendix C. A Dynamic Link Between Two Nodes

This appendix appears in its original form, without editorial change.

An event cascade for link extrusion between two network nodes starts at the <ROUTE>. Commenting is consistent in a d²NetVis scene of MDS-HVI output for an Xj3D view.

```xml
<!-- link from RI to C1 -->

<Group DEF="GROUP_link_RI_C1">
  <Transform DEF="TRANSFORM_link_RI_C1"
             translation="0.0 0.0 0.0">

    <!-- appearance and geometry -->

    <Shape DEF="SHAPE_link_RI_C1">
     <Appearance>
      <!-- appearance of link -->

      <Material DEF="MATERIAL_link_RI_C1"
                diffuseColor="0.0 0.3 1.0"
                transparency="0.75"/>

      <!-- appearance of packet -->

      <PixelTexture image="1 16 4 0xFF4C80FF
                           0x004CFF80 0x004CFF80 0x004CFF80
                           0x004CFF80 0x004CFF80 0x004CFF80
                           0x004CFF80 0x004CFF80 0x004CFF80
                           0x004CFF80 0x004CFF80 0x004CFF80
                           0x004CFF80 0x004CFF80 0xFF4C80FF"/>
      <TextureTransform DEF="TEXTURE_TRANSFORM_linkPacket_1_2"
                        translation="0.0 0.1" scale="1.0 0.8"/>
     </Appearance>

     <Cylinder DEF="CYLINDER_linkPacket_1_2"
               radius="0.015" height="1.0"
               top="false" bottom="false"/>
    </Shape>

    <!-- for toggle of text on the link -->

    <Switch DEF="SWITCH_link_RI_C1">
     <Transform translation="-1.0 0.0 -1.0"
                rotation="0 0 1 0.0">
      <!-- appearance and text; note that      -->
      <!--  Billboard element for rotating text. -->

      <Billboard>
       <Shape>
```

```
          <Appearance>
            <Material USE="MATERIAL_link_RI_C1"/>
          </Appearance>

          <Text DEF="TEXT_link_RI_C1"
                containerField="geometry"
                maxExtent="0.0"
                string="link from RI to C1">
            <FontStyle USE="FONT_STYLE_nodeAndLink"/>
          </Text>
        </Shape>
      </Billboard>
    </Transform>
  </Switch>

    <PlaneSensor DEF="PLANE_SENSOR_link_RI_C1"
                 offset="0.0 0.0 0.0"/>
    <TouchSensor DEF="TOUCH_SENSOR_link_RI_C1"
                 enabled="true"/>
  </Transform>

  <TimeSensor DEF="TIME_SENSOR_link_RI_C1"
              cycleInterval="9.5"
              loop="true"/>
  <PositionInterpolator2D DEF="POSITION_INTERPOLATOR2D_link_RI_C1"
                          key="0.0 1.0"/>
</Group>

<!-- toggle text on the link -->

<Script DEF="SCRIPT_touchSwitch_link_RI_C1">
  <field name="touch_link"       type="SFTime"  accessType="inputOnly"/>
  <field name="onOff_link"       type="SFInt32" accessType="outputOnly"/>
  <field name="active_link"      type="SFBool"
                                 value="false"
                                 accessType="initializeOnly"/>
  <field name="packetKeyValue" type="MFVec2f" accessType="outputOnly"/>
  <field name="packetStart"      type="SFTime"  accessType="outputOnly"/>

  <![CDATA[ecmascript:
                    function touch_link(timestamp)
                    {
//                      if ( active_link ) {
//                        onOff_link  = 1;
//                        active_link = false;
//                      }
```

```
//                              else {
//                                onOff_link  = 0;
//                                active_link = true;
//                               }

                               if ( Math.random() < 0.5 ) {
                                packetKeyValue = new MFVec2f(new SFVec2f(0.0,0.1),new
SFVec2f(0.0,1.4));
                               }
                               else {
                                packetKeyValue = new MFVec2f(new SFVec2f(0.0,1.4),new
SFVec2f(0.0,0.1));
                               }

                               packetStart = timestamp;
                              }
                ]]>
    </Script>

    <!-- dynamic link from RI to C1 -->

    <Script DEF="SCRIPT_link_RI_C1">
     <field name="translationA"        type="SFVec3f"        accessType="inputOnly"/>
     <field name="translationZ"        type="SFVec3f"        accessType="inputOnly"/>
     <field name="spine"               type="MFVec3f"

                                                          value="-0.0739  0.1408  0.0453,
                                                             -0.2006  0.0480  0.0624"
                                                   accessType="initializeOnly"/>
     <field name="translation_changed" type="SFVec3f"        accessType="outputOnly"/>
     <field name="rotation_changed"    type="SFRotation"     accessType="outputOnly"/>
     <field name="scale_changed"       type="SFVec3f"        accessType="outputOnly"/>

     <![CDATA[ecmascript:
                         function calculate()
                         {
                           translation_changed = spine[0].add(spine[1]).divide(2.0);

                           var vec = spine[1].subtract(spine[0]);

                           scale_changed = new SFVec3f(1.0,vec.length(),1.0);

                           var vertical = new SFVec3f(0.0,1.0,0.0);

                           rotation_changed = new SFRotation(vertical,vec);
                         }
```

```
function initialize()
{
  calculate();
}

function translationA(value)
{
 spine[0] = value;

  calculate();
}

function translationZ(value)
{
 spine[1] = value;

  calculate();
}
             ]]>
    </Script>
```

<!-- RI-C1 link extrusion -->

```
<ROUTE fromNode="TRANSFORM_RI"          fromField="translation"
       toNode= "SCRIPT_link_RI_C1"      toField=  "translationA"/>
<ROUTE fromNode="TRANSFORM_C1"          fromField="translation"
       toNode= "SCRIPT_link_RI_C1"      toField=  "translationZ"/>
<ROUTE fromNode="SCRIPT_link_RI_C1"     fromField="translation_changed"
       toNode= "TRANSFORM_link_RI_C1"   toField=  "translation"/>
<ROUTE fromNode="SCRIPT_link_RI_C1"     fromField="rotation_changed"
       toNode= "TRANSFORM_link_RI_C1"   toField=  "rotation"/>
<ROUTE fromNode="SCRIPT_link_RI_C1"     fromField="scale_changed"
       toNode= "TRANSFORM_link_RI_C1"   toField=  "scale"/>
```

INTENTIONALLY LEFT BLANK.

List of Symbols, Abbreviations, and Acronyms

3-D	three dimensional
ARL	U.S. Army Research Laboratory
d^2NetVis	dynamically generated nodes and links for a dynamic network visualization
DAG	directed acyclic graph
ECMAscript	European Computer Manufacturers Association Scripting Language
ISO	International Standards Organization
MDS	multidimensional scaling
MDS-HVI	multidimensional scaling high-value individual
SG	scene graph
X3D	extensible three-dimensional language
Xj3D	extensible three-dimensional viewer with Java bindings, Yumetech, Inc.

NO. OF
COPIES   ORGANIZATION

  1   DEFENSE TECHNICAL
 (PDF   INFORMATION CTR
 only)   DTIC OCA
         8725 JOHN J KINGMAN RD
         STE 0944
         FORT BELVOIR VA 22060-6218

  1   DIRECTOR
         US ARMY RESEARCH LAB
         IMNE ALC HRR
         2800 POWDER MILL RD
         ADELPHI MD 20783-1197

  1   DIRECTOR
         US ARMY RESEARCH LAB
         RDRL CIM L
         2800 POWDER MILL RD
         ADELPHI MD 20783-1197

  1   DIRECTOR
         US ARMY RESEARCH LAB
         RDRL CIM P
         2800 POWDER MILL RD
         ADELPHI MD 20783-1197

  1   DIRECTOR
         US ARMY RESEARCH LAB
         RDRL D
         2800 POWDER MILL RD
         ADELPHI MD 20783-1197

         ABERDEEN PROVING GROUND

  1   DIR USARL
         RDRL CIM G (BLDG 4600)

INTENTIONALLY LEFT BLANK.